8X

УДК 519.6

А.Н. Химич, В.В. Полянко, А.В. Попов, О.В. Рудич
Институт кибернетики им. В.М. Глушкова НАН Украины, г. Киев
dept150@insyg.kiev.ua

Решение задач расчета прочности конструкций на MIMD-компьютере

Рассматривается решение задач расчета прочности конструкций на компьютерах с параллельной организацией вычислений, в том числе алгоритмы формирования, исследования и решения систем разрешающих уравнений, получаемых при дискретизации методом конечных элементов названных задач. Приводятся некоторые результаты численных экспериментов на MIMD-компьютере.

Введение

При проектировании объектов различного назначения часто возникает необходимость в проведении расчетов прочности конструкции таких объектов или отдельных ее элементов. В первую очередь задачи расчета прочности конструкций возникают в различных отраслях машиностроения (судостроении, авиастроении, ракетостроении, моторостроении и т.п.), а также в промышленном и гражданском строительстве (расчет отдельных конструктивных элементов или сооружений в целом).

Для проведения прочностных расчетов создано множество программных средств, как универсального назначения, так и ориентированных на определенную отрасль (NASTRAN, MARC, ANSYS и другие, см. также [1], [2]). Например, для решения задач расчета прочности строительных конструкций разработан программный комплекс (ПК) ЛИРА [2], [3]. Подстроив его интерфейс для соответствующей предметной области, ПК ЛИРА можно использовать для прочностных расчетов в других отраслях.

Возрастающие требования к качеству проектных решений, применение новых конструктивных материалов вызывает необходимость в решении качественно новых задач, а также в выполнении расчетов уникальных конструкций. Растет потребность в новых методах и подходах, связанных с построением и исследованием корректных компьютерных моделей, которые адекватно отображают реальную работу конструкций. Возрастают требования к достоверности получаемых компьютерных результатов. Эти факторы приводят к тому, что существенно возрастает объем обрабатываемой информации, что, в свою очередь, вызывает рост требований к вычислительным ресурсам.

Разрешить эту проблему позволят использование в прочностных расчетах MIMD-компьютеров и интеллектуализации процессов исследования и решения задач. Так, на базе интеллектуальных программ библиотеки Inparlib [4], [5], препроцессора и постпроцессора ПК ЛИРА разработан программный комплекс по исследованию и решению некоторых задач расчета прочности строительных конструкций на интеллектуальных параллельных компьютерах Инпарком [6].

Математические постановки задач

Математически задачи расчета прочности конструкций с использованием принципа возможных перемещений могут быть поставлены в виде следующих вариационных задач [7]: необходимо найти вектор-функцию $u \in U_0$, которая для любой вектор-функции $v \in U_0$ (любого возможного перемещения) удовлетворяет одному из интегральных тождеств:

– для статической задачи

$$\boldsymbol{a}(u,v) = \boldsymbol{l}(f,v); \tag{1}$$

– для динамической задачи

$$\boldsymbol{a}(u,v) + \boldsymbol{b}(u'',v) + \boldsymbol{c}(u',v) = \boldsymbol{l}(f,v), \quad u(t_0) = u^{(0)}, \quad u'(t_0) = u^{(1)}; \tag{2}$$

– для задачи о собственных колебаниях

$$\boldsymbol{a}(u,v) = \lambda \boldsymbol{b}(u,v), \tag{3}$$

где U_0 – бесконечномерное функциональное пространство возможных перемещений, симметричные билинейные функционалы $\boldsymbol{a}(u,v)$, $\boldsymbol{b}(u'',v)$, $\boldsymbol{c}(u',v)$ пропорциональны соответственно потенциальной, кинетической энергиям деформации и энергии торможения, а линейный функционал $\boldsymbol{l}(f,v)$ пропорционален работе приложенных (внешних) усилий при нагружении. Через u' обозначена первая производная вектор-функции $u(t,\chi)$ по времени, а через u'' – вторая.

Здесь рассмотрены только линейные задачи, поскольку предполагается, что решение нелинейной задачи может быть сведено к решению последовательности линейных задач.

Дискретизация задач методом конечных элементов

Теоретической основой большинства программных средств для расчета прочности конструкций является метод конечных элементов (МКЭ) [8], реализованный в форме перемещений. Выбор именно этой формы объясняется простотой ее алгоритмизации и физической интерпретации, наличием единых методов построения матриц жесткости и векторов нагрузок для различных типов конечных элементов, возможностью учета произвольных граничных условий и сложной геометрии рассчитываемой конструкции.

Дискретизация вариационной задачи методом конечных элементов заключается в замене бесконечномерного пространства допустимых функций U_0 его конечномерным подпространством U_0^h кусочно-полиномиальных функций (пространством метода конечных элементов). Для построения такого пространства исходная область разбивается на подобласти-элементы. На каждом элементе функции из пространства МКЭ являются полиномами заданной степени, коэффициенты которых определяются через значение функций (а в ряде случаев и их производных) в узлах элемента (точки в вершинах, на сторонах, гранях или внутри элемента). Таким образом, функции такого пространства МКЭ однозначно определяются конечным числом своих значений (и значений своих производных) в узлах элементов, которые образуют в исходной области задачи, включая границу, конечно-элементную сетку. Размерность пространства МКЭ U_0^h равна общему количеству этих неизвестных значений в узлах сетки (эти значения называют еще узловыми параметрами или степенями свободы).

Узловым параметрам соответствуют базисные (координатные) вектор-функции φ_i, отличные от нуля только на элементах, которые содержат соответствующий данному узловому параметру узел. Для узловых параметров и базисных функций справедливы следующие соотношения

$$L_j(\varphi_i) = \delta_{ij}, \tag{4}$$

где δ_{ij} – символ Кронекера, а результатом операции $L_j(\varphi_i)$ является значение компоненты вектор-функции φ_i для степени свободы (узлового параметра) L_j. Вектор-функции из подпространства U_0^{h} могут быть представлены в виде удовлетворяющей главным (кинетическим) краевым условиям линейной комбинации базисных вектор-функций

$$u^h(\text{ч}) = \sum_{i=1}^{n} x_i \text{ц}_i(\text{ч}), \tag{5}$$

где φ_i ($i = 1, 2, \ldots, n$) – упомянутый выше кусочно-полиномиальный базис U_0^{h}.

Тогда билинейные и линейный функционалы из (1) – (3) в конечномерном пространстве U_0^{h} можно записать в виде соответственно билинейных и линейной форм узловых параметров:

$$\boldsymbol{a}(u^h,v^h) \equiv y^{\mathsf{T}}Ax, \quad \boldsymbol{b}(u^h,v^h) \equiv y^{\mathsf{T}}Bx, \quad \boldsymbol{c}(u^h,v^h) \equiv y^{\mathsf{T}}Cx, \quad \boldsymbol{l}(f,v^h) \equiv y^{\mathsf{T}}b, \tag{6}$$

где x и y – векторы узловых параметров соответственно функций u^h и v^h, а элементы матриц жесткости (A), масс (B), демпфирования (C) и вектора нагрузок (b) вычисляются по формулам ($i, j = 1, 2, \ldots, n$)

$$a_{ij} = \boldsymbol{a}(\varphi_i, \varphi_j), \quad b_{ij} = \boldsymbol{b}(\varphi_i, \varphi_j), \quad c_{ij} = \boldsymbol{c}(\varphi_i, \varphi_j), \quad b_i = \boldsymbol{l}(f, \varphi_i). \tag{7}$$

Учитывая, что базисные функции подпространства МКЭ U_0^{h} можно выбрать так, чтобы они были отличными от нуля только на нескольких конечных элементах, структура матриц из (6) является разреженной и определяется нумерацией узловых параметров, которая, в свою очередь, зависит от нумерации узлов конечноэлементной сетки (матрица называется разреженной, если число ее элементов, отличных от нуля, намного меньше общего их числа – n^2).

Таким образом, учитывая в (6) произвольность вектора y, разрешающие дискретные задачи имеют вид [9]:

– системы линейных алгебраических уравнений (СЛАУ)

$$Ax = b \tag{8}$$

для статической задачи (1);

– задачи с начальными условиями

$$Bx''(t) + Cx'(t) + Ax(t) = b(t), \quad x(t_0) = x^{(0)}, \quad x'(t_0) = x^{(1)} \tag{9}$$

для динамической задачи (2);

– обобщенной алгебраической проблемы собственных значений

$$Ax = \lambda^{\mathrm{h}}Bx \tag{10}$$

для задачи о собственных колебаниях (3).

Достоверность приближенных решений

Когда речь идет о достоверности решения, то имеется в виду, что исследуется достоверность приближенных решений математических задач (1) – (3). Полная погрешность приближенного решения учитывает влияние погрешностей исходных

данных, погрешностей дискретизации и формирования данных соответствующей дискретной задачи (8), (9) или (10) и погрешностей компьютерного решения этой дискретной задачи.

Особое внимание следует уделить изучению ***влияния приближенных исходных данных*** прикладной задачи. Ведь в реальных задачах расчета прочности конструкций почти все исходные данные – константы, которыми задаются механические свойства материала конструкции (например, модуль Юнга), геометрические размеры, нагрузки и тому подобное, как правило, задаются с погрешностями. Для учета влияния погрешностей в исходных данных на решения задач теории упругости (поведение наследственной погрешности) необходимо исследовать устойчивость решения задачи по коэффициентам уравнений и правой части. Задача называется коэффициентно устойчивой, если при малом возмущении коэффициентов задачи ее решение также меняется мало. В [10] рассмотрен вопрос коэффициентной устойчивости для первой основной задачи теории упругости (когда на границе области заданы напряжения). Описанный в этой работе метод исследования коэффициентной устойчивости первой основной краевой задачи теории упругости можно использовать для других задач теории упругости и для исследования устойчивости по правой части.

Для МКЭ в перемещениях известны условия ***сходимости и оценки погрешности решения*** [3], [8], [9]. Необходимыми условиями сходимости решения, полученного с помощью МКЭ, являются выполнение условий существования и единственности решения, линейная независимость и полнота системы базисных функций, а также их совместность (конформность) либо выполнение условий, компенсирующих несовместность. Линейная независимость следует из равенств (4). Совместность означает, что все базисные функции являются возможными перемещениями, т.е. принадлежат пространству U_0.

Известны легко проверяемые условия, позволяющие установить полноту базисных функций, их совместность или выполнение условий, компенсирующих несовместность. Эти условия имеют вид равенств, которым должны удовлетворять базисные функции на каждом конечном элементе.

При дополнительных условиях на гладкость решений задач (1) – (3) точность метода конечных элементов характеризуется априорной оценкой $\|u - u^h\| \le C_1 h^k$, где h – наибольший из линейных размеров конечных элементов, k – величина, зависящая от степени кусочно-полиномиальных базисных функций, которую называют порядком сходимости, C_1 – постоянная, зависящая от гладкости решения задачи.

К сожалению, приведенная оценка имеет асимптотический характер, выполняясь для достаточно малого h, и не является конструктивной из-за неопределенной константы C_1. Однако эта оценка позволяет выбрать для решения конкретной задачи конечноэлементный базис. Этот выбор обусловливается математической моделью, гладкостью решения исходной задачи, необходимой точностью решения, скоростью сходимости МКЭ, а также свойствами разрешающих задач (структурой и порядком матриц, числом обусловленности матрицы), точностью задания исходных данных. Следует также учитывать, что сколь угодно высокая (допускаемая свойствами задачи) точность конечноэлементного решения, как правило, является дорогой, да и не всегда необходимой.

Таким образом, теоретические основы МКЭ позволяют не только исследовать корректность применения известных конечных элементов, но также разработать принципы конструирования новых совместных или несовместных элементов и получить для них оценки погрешности [11].

Существенное влияние на точность решения дискретных задач (8) – (10) могут иметь ***погрешности***, которые возникают в результате использования ***квадратурных формул*** при вычислении (7) элементов матриц (жесткостей, масс, демпфирования) и вектора нагрузок соответствующей задачи. Чаще всего используются квадратурные формулы Гаусса, которые обеспечивают наивысшую алгебраическую точность при одном и том же количестве узлов квадратур и для которых справедлива следующая оценка остаточного члена [12]:

$$|R| \le \frac{(n!)^4}{[(2n)!]^3(2n+1)} C_2 (b-a)^{2n+1},$$

где $[a,b]$ – отрезок интегрирования, n – количество узлов квадратуры. Важно так выбирать количество узлов для численного интегрирования, чтобы его точность по h была не меньшей точности МКЭ. Кроме этого нужно учитывать, что постоянная C_2 равна максимальному значению на отрезке $[a,b]$ абсолютной величины производной $(2n)$-го порядка подынтегральной функции. Поэтому абсолютное значение погрешности интегрирования может в некоторых случаях существенно превысить абсолютное значение погрешности МКЭ, например, если коэффициенты исходных задач значительно изменяются в пределах какого-то конечного элемента. В таком случае следует увеличивать количество узлов интегрирования (на 1 – 2 в каждом координатном направлении).

Погрешность компьютерного решения $\tilde{x}$ относительно точного решения x соответствующей дискретной задачи (8), (9) или (10) зависит от точности задания исходных данных, метода и алгоритма решения этой дискретной задачи, а также погрешностей округления. Так, полная погрешность решения СЛАУ (8) в случае положительно определенной матрицы жесткости A оценивается следующим образом [13]

$$\frac{\|x-\tilde{x}\|}{\|x\|} \le \frac{h(A)}{1-h(A)\varepsilon_A}\left(\varepsilon_A + \varepsilon_b + \frac{\|r\|}{\|b\|}\right),$$

где $r = b - A\tilde{x}$ – невязка, $h(A)$ – число обусловленности матрицы A или его оценка, $\varepsilon_A = \varepsilon_B = masheps$ – относительные погрешности задания в компьютере элементов матрицы и правых частей системы, *masheps* – наименьшее положительное число с плавающей запятой в компьютере, для которого выполняется соотношение $1 + masheps > 1$.

В работе [14] рассмотрено получение с помощью МКЭ достоверного приближения к *единственному решению задачи на подпространстве* (например, в случае задания на границе области напряжений).

После получения решения задачи необходимо провести апостериорный анализ результатов с точки зрения удовлетворения тем гипотезам, которые были использованы при построении математической модели задачи, а также требованиям точности решения. Если решение удовлетворяет всем поставленным требованиям, то можно считать, что этому решению можно доверять и использовать для проектирования и тому подобное.

Формирование и решение дискретных задач

С точки зрения компьютерного решения разрешающих задач (8) – (10) существенное значение имеет ряд их особенностей:

– порядок матриц разрешающих дискретных задач составляет $O(10^5)$ – $O(10^7)$;

– матрицы дискретных задач являются *симметричными*, *положительно определенными* или *положительно полуопределенными*;
– матрицы дискретных задач имеют *разреженную структуру* (ленточную, профильную, блочную и т.п.).

С целью минимизации количества арифметических операций при решении систем линейных алгебраических уравнений, а также требуемой оперативной и внешней памяти, учитывая разреженную структуру матрицы жесткости A задач (8) – (10), в большинстве случаев проводится переупорядочение неизвестных задачи. В зависимости от критерия минимизации – ширины ленты матрицы, ее профиля, общего количества арифметических операций при треугольном разложении матрицы и т. д. – используются различные методы переупорядочения [14], например, обратный алгоритм Катхилла-Макки, фактор деревьев, минимальной степени. Конкретных рекомендаций для выбора метода упорядочения дать нельзя, так как эффективность того или иного алгоритма существенно зависит от первоначальной структуры конкретной матрицы A.

Формирование дискретных задач. Важным преимуществом МКЭ является то, что элементы матриц и векторов правых частей задач (8) – (10) получаются суммированием соответствующих элементов матриц (жесткости, масс, демпфирования) и векторов нагрузок, построенных для отдельных конечных элементов. Такое свойство матриц и векторов дискретных задач МКЭ позволяет эффективно распараллелить процесс формирования этих матриц и векторов.

Например, (*метод декомпозиции области*) область может быть разбита на приблизительно равные подобласти (по числу используемых процессов на основе требований их равномерной загрузки) и для конечных элементов, содержащихся в этих подобластях, параллельно и независимо друг от друга формируются соответствующие матрицы и векторы. После этого с использованием обменов между процессами формируются глобальные матрицы и векторы соответствующей дискретной задачи (8) – (10). При этом глобальные матрицы и векторы могут быть сформированы или в файле (файлах) на внешнем носителе, или распределенными между процессами в соответствии с требованиями метода, который будет использоваться для решения дискретной задачи.

Если нумерация узлов и узловых параметров выполняется так, что узлы, расположенные на общих границах подобластей (внутренние граничные узлы), получают последние номера при сплошной нумерации узлов всей сеточной области, то такой подход позволяет сформировать блочно-диагональную глобальную матрицу с окаймлением. В этом случае эффективность распараллеливания близка к единице, так как обмены между процессами необходимы только при обработке конечных элементов, которые содержат внутренние граничные узлы.

Во многих параллельных алгоритмах решения задач (8) – (10) используется циклическое распределение строк или столбцов матрицы между параллельными процессами. Тогда циклический алгоритм формирования глобальных матриц МКЭ описывается следующим образом: каждый процесс выполняет в цикле для всех групп конечных элементов такие действия:

(i) вычисление всех локальных матриц и векторов подгруппы конечных элементов, которая обрабатывается данным процессом на данном шаге цикла;

(ii) в цикле для $i = 0, 1 ..., p{-}1$ проводится (p – количество процессов):

– рассылка из процесса с логическим номером i всех вычисленных локальных матриц и векторов;
– ассемблирование элементов глобальных матриц и векторов, которые распределяются данному процессу, используя локальные матрицы и векторы текущей группы конечных элементов, которые сформированы процессом с логическим номером i.

Этот алгоритм является наиболее эффективным при циклических схемах распределения матриц и векторов между процессами.

Решение систем линейных алгебраических уравнений. Для решения СЛАУ (8) – исходной или переупорядоченной – на MIMD-компьютере могут быть использованы параллельные алгоритмы для исследования и решения СЛАУ с симметричными матрицами, использующие LDL^T-разложение метода Холецкого [15]. В зависимости от ширины и заполненности ленты матрицы используются:

– блочные параллельные алгоритмы для узких ленточных матриц и блочно-диагональных матриц с окаймлением,

– одномерные блочные циклические параллельные алгоритмы для исследования и решения СЛАУ с ленточными или с матрицами профильными матрицами.

При выборе параллельного алгоритма необходимо учитывать его эффективность. Например, коэффициент эффективности блочного параллельного алгоритма LDL^T-разложение метода Холецкого для узкой ленточной симметричной матрицы оценивается так:

$$E_\text{р} \approx \left(0{,}25 + \frac{1}{8m} + \frac{p-1}{16n}\left(\frac{7p-18}{3}m + \frac{p}{2}\tau_1\right)\right)^{-1},$$

а одномерного блочного циклического параллельного алгоритма LDL^T-разложение метода Холецкого для ленточной симметричной матрицы так:

$$E_p \approx \left(1 + \frac{2(p-1)(s+1)}{m} + \frac{p\log_2 p}{m}\tau_2\right)^{-1},$$

где m – полуширина ленты матрицы, s – количество строк в блоке, $\tau_1 = \tau_\text{o} + \frac{2}{m^2}\tau_\text{c}$, $\tau_2 = \tau_o + \frac{1}{sm}\tau_c$, τ_c, τ_o – отношения времен, необходимых соответственно для установления связей между двумя процессами и для обмена одним машинным словом между двумя процессами, к среднему времени выполнения одной арифметической операции.

Решение задачи с начальными условиями. В полудискретном МКЭ приближенное решение ищется в виде (5), где коэффициенты x_i являются функциями времени t. В результате получается система обыкновенных дифференциальных уравнений второго порядка с начальными условиями (9), где: $x(t)$, $x^{(0)}$, $x^{(1)}$ – векторы с элементами $x_i(t)$, $x_i^{(0)} = L_i\left(u^{(0)}\right)$, $x_i^{(1)} = L_i\left(u^{(1)}\right)$.

Система (9) решается методом разложения по формам собственных колебаний. Если λ_k^h, z_k^h $\left(z_k^{h\text{T}} B z_k^h = 1,\ k = 1, 2, \ldots, n\right)$ – решение алгебраической проблемы собственных значений (10), то, полагая в (9) $x(t) = \sum_{k=1}^{n} y_k(t) z_k^h$, получим вместо (10) (при B-ортогональности векторов z_k^h и определенных предположениях относительно матрицы демпфирования C) систему, которая распадается на независимые уравнения относительно $y_i(t)$:

$$\begin{aligned} & y_k''(t) + 2\xi_k \omega_k y_k'(t) + \omega_k^2 y_k(t) = P_k(t), \quad t > t_0, \\ & y_k(t_0) = y_k^{(0)}, \quad y_k'(t_0) = y_k^{(1)}, \end{aligned} \tag{11}$$

где $\omega_i = \lambda_i^{-0{,}5}$, $0 < \xi_k < 1$, $P_k(t) = z_k^{h\text{T}} b(t)$, $y_k^{(0)} = x^{(0)\text{T}} B z_k^h$, $y_k^{(1)} = x^{(1)\text{T}} B z_k^h$, $k = 1, 2, \ldots, n$.

Решения задач (11) имеют вид:

$$y_k(t) = e^{-\xi_k \omega_k t}\left(\frac{y_k^{(1)} + y_k^{(0)}\xi_k\omega_k}{w_k}\sin\overline{\omega}_k t + y_k^{(0)}\cos\overline{\omega}_k t\right) +$$
$$+\frac{1}{\overline{\omega}_k}\int_{t_0}^{t} P_k(\tau)^{-\xi_k\omega_k(t-\tau)}\sin\overline{\omega}_k(t-\tau)d\tau,$$

где $\overline{\omega}_k = \omega_k\sqrt{1-\xi_k}$. Анализ этих решений свидетельствует о том, что существенный вклад в решение $x(t)$ задачи (9) вносят лишь около 10 составляющих, соответствующих минимальным собственным значениям. Поэтому соответствующая задача на собственные значения вида (10) может решаться методом итераций на подпространстве [16].

Существует ряд случаев нагружений, когда возможно точное вычисление решения задачи (11). Например, для ветровой, сейсмической или гармонической нагрузок. В остальных случаях решение $y_k(t)$ находится численно, например, методом конечных разностей по схеме Ньюмарка. Также задачи Коши (11) на MIMD-компьютере могут решается методом Рунге-Кутта четвертого порядка, ориентированным на решение систем обыкновенных дифференциальных уравнений второго порядка.

Решение обобщенной алгебраической проблемы собственных значений. Решение на MIMD-компьютере обобщенной алгебраической проблемы собственных значений (10) с ленточными или профильными симметричными матрицами (исходными или переупорядоченными) может быть получено с помощью параллельных алгоритмов метода итераций на подпространстве, которые, как и в случае решения СЛАУ (8), используют одномерное блочно-циклическое распределение данных между процессами [17].

Решение задач расчета строительных конструкций на Инпарком

Как отмечалось выше, для исследования и решения задач (1) – (3) может быть использован программный комплекс (ПК) по решению некоторых задач расчета прочности строительных конструкций на многопроцессорных рабочих станциях Инпарком. Ниже представлена технологическая схема этого программного комплекса (табл. 1).

Таблица 1 – Технологическая схема программного комплекса Инпарком

Создание КЭ-модели конструкции	Формирование данных разрешающей задачи	Решение разрешающей задачи	Обработка результатов решения разрешающей задачи	Анализ результатов расчета
интерактивно на языке предметной области	автоматически	автоматически	автоматически	интерактивно
препроцессор ПК ЛИРА	препроцессор ПК ЛИРА 9.2	Inparlib	препроцессор ПК ЛИРА 9.2	препроцессор ПК ЛИРА
локально на хост-компьютере Инпарком или ***удаленно***, используя Интернет	***локально*** на хост-компьютере Инпарком	***локально*** на обрабатывающем блоке Инпарком	***локально*** на хост-компьютере Инпарком	***локально*** на хост-компьютере Инпарком или ***удаленно***, используя Интернет

В этом случае сформированные на графической станции препроцессором программного комплекса исходные данные соответствующей дискретной задачи (8) – (10) записываются в бинарные файлы, используя принятые в ПК ЛИРА форматы данных. После этого с помощью специального интерфейса на обрабатывающие узлы компьютера Инпарком загружается параллельная программа решения соответствующей дискретной задачи, которая:

– анализируя параметры задачи, определяет алгоритм решения задачи и параметры, необходимые для решения;

– считывает исходные данные из файлов и распределяет их между процессами в соответствии с выбранным параллельным алгоритмом решения задачи;

– исследует и решает задачу, обращаясь к соответствующим функциональным модулям библиотеки Inparlib [4], [5];

– сохраняет результаты исследования и решения задачи в файлах для дальнейшей обработки и использования постпроцессором программного комплекса.

Результатами статического расчета напряженно-деформированного состояния являются перемещения узлов схемы и усилия (напряжения) в сечениях элементов. Результатами динамического расчета – решения АПСЗ (10) – являются периоды, частоты и формы собственных колебаний для каждого тона. Результатами решения задачи Коши (11) являются значения перемещений $y_{k,j} = y_k(t_j)$ в узлах.

Программным комплексом по решению некоторых задач расчета прочности строительных конструкций на рабочих станциях Инпарком проводилось решение ряда тестовых статических задач, для которых разрешающими являются СЛАУ (8). В табл. 2 приведены времена решения СЛАУ тестовых задач на рабочих станциях Инпарком, а также ориентировочные времена решения этих задач на графической станции компьютера Инпарком программным комплексом ЛИРА 9.2.

Следует обратить внимание, что проводимые функциональными модулями Inparlib исследования увеличивают общее время решения задачи, как правило, не более, чем на 10 %. Однако такие исследования (особенно вычисление оценки числа обусловленности матрицы СЛАУ и сопоставление этого значения с погрешностями исходных данных задачи при вычислении оценки наследственной погрешности) позволяют оценить достоверность полученного решения и, что особенно важно, внести коррективы в постановку задачи при невозможности гарантировать достоверность решения задачи.

Таблица 2 – Времена решения СЛАУ тестовых задач

№ п/п	Порядок системы	Полу-ширина (макс.) ленты	Плот-ность	Количе-ство процессов	Время решения (мин.)		Отношение времен решения
					на Инпарком	ПК Лира 9.2	
1	44 436	4 476	21 %	64	0,200	1,91	9,55
				96	0,183		10,42
2	44 436	37 580	2 %	16	0,117	0,76	6,51
				80	0,083		9,12
3	189 956	22 585	9 %	16	1,937	35,58	18,37
4	283 031	19 530	7 %	16	1,222	16,40	23,42
5	661 590	34 242	5 %	16	10,083	54,24	5,38
				96	5,200		10,43
6	1 000 332	1 004	85 %	16	3,001	41,00	13,66
7	1 200 000	1 265	100 %	16	4,776	102,41	21,36

Пример расчета на прочность конструкции здания (рис. 1) с помощью программного комплекса по решению некоторых задач расчета прочности строительных конструкций на рабочих станциях Инпарком.

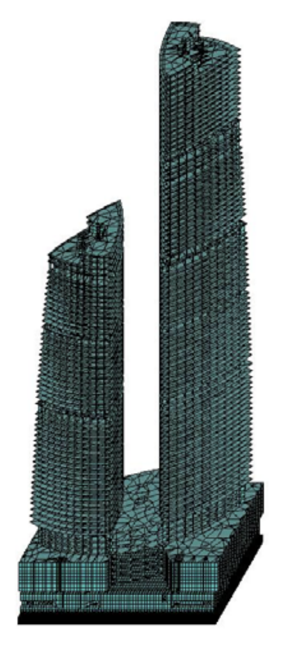


Рисунок 1 – Конструкция здания для примера расчета на прочность

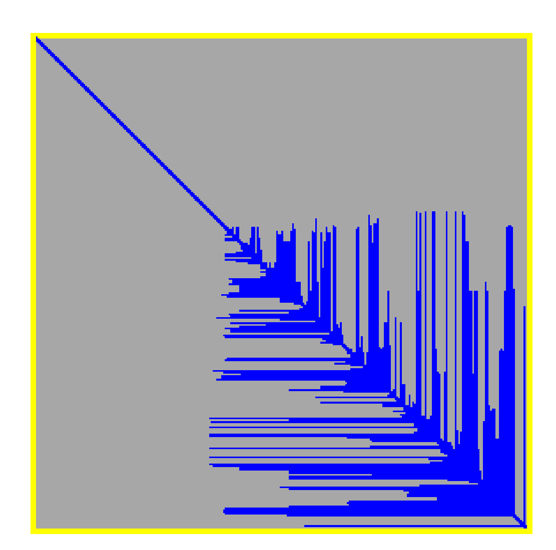


Рисунок 2 – Структура переупорядоченной матрицы

Конструкция разбита на 972 808 конечных элементов. Образованная в результате разбиения конечно-элементная сетка содержит 895 302 узла.

В результате дискретизации получена разрешающая задача (8) – система линейных алгебраических уравнений, порядок которой равен 5 371 727. После оптимизации структуры матрицы (переупорядочения) методом минимальной степени плотность (заполненность ленты матрицы ненулевыми элементами) переупорядоченной структуры составила менее 1 %, хотя полуширина ленты стала равной 3 174 975. Структура переупорядоченной матрицы представлена на рис. 2. Размер файла для хранения информации о структуре матрицы и значений ее ненулевых элементов составляет около 76 Gb.

Задача была сформирована и решалась с помощью программного комплекса на рабочих станциях Инпарком. Для решения разрешающей системы (8) использовались функциональные модули библиотеки Inparlib. Расчеты проводились для одного случая нагружения. Для расчетов использовалось различное количество процессоров (и процессов) рабочих станций Инпарком-64 и Инпарком 128. В табл. 3 приведены времена решения дискретной задачи.

Таблица 3 – Времена решения дискретной задачи

Компьютер	Инпарком-128			Инпарком-64		Графическая станция
Количество процессоров	32	24	20	16	12	1
Количество процессов	128	96	80	64	48	1
Время решения СЛАУ	6:09	7:27	8:18	9:24	10:22	115 часов (оценка)

Выводы

Проведенные авторами исследования и апробация программного комплекса по решению некоторых задач расчета прочности строительных конструкций на рабочих станциях Инпарком показали перспективность создания программных средств по реше-

нию прикладных задач на параллельных компьютерах путем встраивания параллельных программ по решению отдельных подзадач в существующие программные комплексы для решения прикладных задач в различных отраслях науки и техники. В качестве таких параллельных программ могут использоваться многие программы библиотеки Inparlib или ее аналогов на других компьютерах с параллельной обработкой информации.

Также является перспективным расширение круга решаемых задач с помощью программного комплекса по решению некоторых задач расчета прочности строительных конструкций на рабочих станциях Инпарком и распараллеливание формирования данных дискретной задачи и обработки результатов решения этой задачи, используя предложенные в настоящей работе алгоритмы.

Литература

1. Исследование с помощью многопроцессорного вычислительного комплекса распределенных систем с большими объемами связанных данных / Сергиенко И.В., Дейнека В.С., Химич А.Н., Попов А.В., Яковлев М.Ф., Калынюк Н.А., Чистякова Т.В., Белоус М.В. – Киев, 2005. – Препр./НАН Украины. Ин-т кибернетики им. В.М. Глушкова; 2005-1. – 33 с.
2. Режим доступа: http://www.lira.com.ua.
3. Городецкий А.С., Евзеров И.Д. Компьютерные модели конструкций. – Киев: ФАКТ, 2005. – 344 с.
4. Численное программное обеспечение интеллектуального MIMD-компьютера Инпарком / Химич А.Н., Молчанов И.Н., Мова В.И. и др. – Київ: Наукова думка, 2007. – 216 с.
5. Герасимова Т.А., Зубатенко В.С., Молчанов И.Н. и др. Библиотека интеллектуальных параллельных программ для исследования и решения задач вычислительной математики с приближенно заданными исходными данными. Свідоцтво на реєстрацію авторського права на твір № 17213 від 11.07.2006 р. / Держ. департамент. інтелект. власності.
6. Режим доступа: http://www.inparcom.com.
7. Тимошенко С.П., Гудьер Дж. Теория упругости. – М.: Наука, 1975. – 575 с.
8. Стренг Г., Фикс Дж. Теория метода конечных элементов. – М.: Мир, 1973. – 349 с.
9. Молчанов И.Н., Николенко Л.Д. Основы метода конечных элементов. – Київ: Наукова думка, 1984. – 267 с.
10. Галба Е.Ф., Гладкий А.В., Химич А.Н., Яковлев М.Ф. О корректности первой основной задачи теории упругости на подпространстве // Компьютерная математика. – 2002. – № 1.– С. 54-62.
11. Молчанов И.Н., Попов А.В. Схемы повышенного порядка точности для некоторых задач на собственные значения // Вариационно-разностные методы в математической физике. Сборник научных трудов. – Москва, 1984. – С. 183-195
12. Бахвалов Н.С. Численные методы. – М.: Наука. – 1973. – Т. 1. – 632 с.
13. Химич А.Н. Оценки полной погрешности решения систем линейных алгебраических уравнений для матриц произвольного ранга // Компьютерная математика. – 2002. – № 2. – С. 41-49.
14. Джордж А., Лю Дж. Численное решение больших разреженных систем уравнений. – М.: Мир, 1984. – 333 с.
15. Попов А.В. Параллельные алгоритмы решения линейных систем с разреженными симметричными матрицами // Проблеми програмування, // Матеріали 6-ї міжнар. науково-практичної конф. з програмування УкрПРОГ'2008. – 2008 р. – №2-3. – С.111-118.
16. Молчанов И.Н., Попов А.В., Химич А.Н. Алгоритм решения частичной проблемы собственных значений для больших профильных матриц // Кибернетика и системный анализ. – 1992. – № 2. – С. 141-147.
17. Зубатенко В.С., Майстренко А.С., Молчанов И.Н. и др. Исследование некоторых параллельных алгоритмов решения задач линейной алгебры на MIMD-компьютерах // Искусственный интеллект. – 2006. – № 3. – С. 129-138.

О.М. Хіміч, В.В. Полянко, О.В. Попов, О.В. Рудич
Розв'язування задач розрахунку міцності конструкцій на MIMD-комп'ютері
Розглядаються блочні алгоритми дослідження та розв'язування задач лінійної алгебри з розрідженими (вузькими стрічковими, з обрамленням і т.п.) симетричними матрицями на комп'ютерах MIMD-архітектури. Досліджується ефективність даних алгоритмів. Наводяться деякі результати чисельних експериментів на MIMD-комп'ютері.

A.N. Khimich, V.V. Polyanko, A.V. Popov, O.V. Rudich
The Solving of Problems of Structural Analysis on MIMD-computer
The solving of problems of structural analysis on parallel computers are dealt with, including algorithms of forming, investigating and solving of systems of the settling equations received during discretization by the finite elements method. Some results of numeral tests carried out on MIMD-computer are given.

Статья поступила в редакцию 18.07.2008.